

## Cloud Robotics: Auto Rover 4WD y Cuadricóptero controlados remotamente desde AWS

Armando De Giusti<sup>1,2</sup>, Ismael Rodriguez<sup>1</sup>, Manuel Costanzo, Marcos Boggia

<sup>1</sup> Instituto de Investigación en Informática LIDI (III-LIDI), Facultad de Informática, Universidad Nacional de La Plata – Comisión de Investigaciones Científicas de la Provincia de Buenos Aires, Argentina.

<sup>2</sup> Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET) Argentina

{degiusti, ismael} @lidi.info.unlp.edu.ar, manuelcostanzo22@gmail.com, marcos\_boggia@hotmail.com

**Resumen.** Este trabajo presenta el despliegue de un sistema multi-robot, integrado por un chasis de auto Rover de 4 ruedas y un cuadricóptero Parrot Bebop, conectados al Cloud público de Amazon Web Services. Se detallan los prototipos desarrollados, como así también el protocolo de comunicación utilizado con el servicio de AWS IoT, y los algoritmos implementados para simular el control remoto de un vehículo de 4 ruedas y un dispositivo de vuelo, ambos, no tripulados.

**Palabras Clave:** Cloud Robotics, Cloud Computing, Internet of Things, AWS, MQTT, NODE-RED, Cuadricóptero, Drone, Rover 4WD, IoT.

### 1 Introducción

Los avances en el paradigma de Cloud Computing han provocado un factor disruptivo de las TI en la industria tecnológica. Según el Instituto Nacional de Estándares y Tecnologías del Departamento de Comercio de los EEUU (NIST), como en varias publicaciones de diversos autores, se ha definido a Cloud Computing como: “*un paradigma informático de cómputo distribuido, que proporciona grandes conjuntos de recursos virtuales (como ser hardware, plataformas de desarrollo, almacenamiento y/o aplicaciones), fácilmente accesibles y utilizables por medio de una interfaz de administración web. Estos recursos son proporcionados como servicios (del inglés, “as a service”) y pueden ser dinámicamente reconfigurados para adaptarse a una carga de trabajo variable (escalabilidad), logrando una mejor utilización y evitando el sobre o sub dimensionamiento (elasticidad). El acceso a los recursos se realiza bajo la demanda de los usuarios, en base a un modelo de autoservicio*” [1].

El modelo de Cloud Computing presenta las características y beneficios siguientes: recursos disponibles bajo demanda, escalabilidad y elasticidad, aprovisionamiento automático de recursos y autoservicio.

Dicho paradigma, brinda al menos tres modelos de despliegue: Cloud Público, Cloud Privado y Cloud Híbrido [2] [3].

Por otro lado, considerando la gran influencia de la robótica en la sociedad y la diversidad de servicios ofrecidos por robots, nos encontramos que los mismos presentan grandes limitaciones en consumo de energía, poder de cómputo, capacidad de almacenamiento, toma de decisiones, tareas cognitivas, etc.

En el año 2010, comenzaron a surgir proyectos de investigación (ej.: RoboEarth [4]), que integraban las tecnologías de Cloud con los sistemas de robots. Es así, que James Kuffner propone el concepto de Cloud Robotics, basado en combinar las tecnologías de robots con el paradigma de Cloud Computing [5].

La idea de Cloud Robotics permite, por medio de aplicaciones, tratar los datos de los componentes de hardware del robot (sensores, actuadores, cámaras, microcontroladores, memoria, etc.), sin importar las limitaciones de cómputo de las placas de desarrollo con microcontroladores y la capacidad de almacenamiento de las mismas [6]. En otras palabras, este concepto permite a los robots obtener resultados de tareas de cómputo intensivo, tales como: procesamiento de imágenes, reconocimiento de voz, determinación de rutas, confección de mapas, acciones cognitivas, etc., sin tratamiento local, sino en el Cloud.

Este paradigma brinda la capacidad de establecer escenarios para sistemas de multi-robot, donde cada robot se integra de un hardware mínimo; una placa microcontroladora con conectividad WiFi permite al robot comunicarse con el Cloud y otros robots. Los datos de sensores y/o sistemas de adquisición de imágenes se procesan en el Cloud y los actuadores de cada robot llevarán a cabo las operaciones necesarias [7].

El propósito de investigación del presente trabajo es el despliegue de un sistema de multi-robot, inicialmente con dos robots conectados al Cloud Público de Amazon Web Services con el fin de simular el control remoto de un vehículo de 4 ruedas y un dispositivo de vuelo, ambos, no tripulados. Para alcanzar tales objetivos, se determinó el método de comunicación con el Cloud; se registraron los robots como dispositivos o “cosas” en el servicio de AWS IoT [8]; también, se confeccionaron los prototipos de cada robot y se desarrollaron los algoritmos que permiten controlar las acciones a ejecutar por cada robot.

El presente trabajo está estructurado de la manera siguiente: en la Sección 2, se introducen algunos conceptos elementales. A continuación, la Sección 3, presenta los prototipos confeccionados y sus componentes. En la Sección 4, se describe el trabajo experimental realizado y los resultados obtenidos. Por último, la Sección 5, expone las líneas de trabajo futuro en relación a esta investigación.

## **2 Algunos conceptos elementales**

En esta Sección, se mencionan algunos conceptos empleados en la investigación realizada.

#### ▪ Amazon Web Services

Es una plataforma de Cloud Público, que provee “Infraestructura como servicio”; a través de la tecnología de virtualización, brinda un gran conjunto de recursos de cómputo, como almacenamiento y capacidad de procesamiento, que pueden ser solicitados a demanda, como así también, se adecuan dinámicamente en tamaño conforme la necesidad del consumidor. Los servicios más destacados de AWS son “Elastic Compute Cloud” (EC2), “Simple Storage Service” (S3) y en nuestro caso utilizaremos los servicios de “AWS Internet of Things” (AWS IoT) [9].

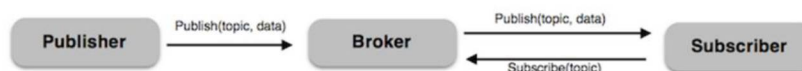
#### ▪ Internet of Things (IoT)

Es un nuevo paradigma cuya definición deriva de considerar “objetos” o “cosas” conectadas a Internet por medio de redes Wireless, sean estas de tecnología WiFi o 4G LTE, utilizando protocolos de comunicación estándar. Dichas “cosas” pueden considerarse etiquetas de identificación por radio frecuencia (RFID), sensores, actuadores, teléfonos móviles, placas de desarrollos, etc. [10].

#### ▪ Protocolo MQTT

Message Queuing Telemetry Transport [11] es un protocolo de comunicación ligero, especialmente diseñado para tolerar conexiones intermitentes y reducir los requisitos de ancho de banda de la red. Desde finales del año 2014, fue presentado como un estándar abierto OASIS [12] y se ha convertido en el protocolo por excelencia para IoT. Soporta comunicación segura con TLS. Maneja tres niveles de calidad de servicio: QoS 0: A lo sumo una vez la entrega del mensaje. QoS 1: Al menos una vez la entrega del mensaje. QoS 2: Exactamente una vez la entrega del mensaje.

Este protocolo, implementa la comunicación de mensajes por medio de la publicación/suscripción sobre un tópico específico (canal de comunicación), como se puede observar en la figura siguiente:



*Figura 1: MQTT: publicación/suscripción.*

#### ▪ AWS IoT

Es un servicio que proporciona AWS con el fin de conectar, administrar y operar grandes conjuntos de dispositivos o cosas. Ofrece mecanismos de conectividad y seguridad para la transmisión de datos. También, permite que los datos, una vez enviados a la plataforma, puedan ser procesados por las aplicaciones de análisis masivo de datos (Elastic MapReduce), análisis predictivo para aprendizaje automático (Amazon Machine Learning), almacenamiento en bases de datos, etc. [13].

AWS IoT permite conectar fácilmente dispositivos al Cloud y a otros dispositivos. El servicio es compatible con los protocolos: HTTP, WebSockets y

MQTT. Amazon ha optado por este último, por ende AWS IoT utiliza la especificación del protocolo MQTT v.3.1.1 con QoS 0 y 1.

Cada dispositivo debe ser registrado como una “cosa” en AWS IoT, para lo cual se emitirá un certificado y un par de llaves privada-pública para el mismo. El certificado y la llave privada, junto con el certificado de la Entidad Certificantes (CA) de Amazon, deberán ser instalados en el dispositivo con el fin que este pueda conectarse al servicio de AWS IoT previa autenticación necesaria.

#### ▪ **Node-RED**

Es una herramienta de programación visual, que permite programar algoritmos basados en flujos para IoT, sin la necesidad de escribir código. Provee de un editor de flujo basado en navegador WEB, que brinda una amplia paleta de nodos con diversa funcionalidad. El flujo se confecciona conectando los nodos entre sí. La funcionalidad de cada nodo está desarrollada sobre Node.js. Los flujos creados son almacenados utilizando JSON, lo cual facilita la importación o exportación para compartir los mismos [14].

### **3 Prototipos y componentes**

Con el fin de simular el control remoto de un vehículo de 4 ruedas y un dispositivo de vuelo, no tripulados, se han implementado dos prototipos de robots que interactúan con el Cloud público de AWS, vía conexión WiFi y Ethernet; de aquí en adelante se denomina al primer robot, como “Rover 4WD” y al segundo, “Cuadricóptero”.

Asimismo, se ha desplegado una instancia de AWS EC2, para ejecutar los algoritmos que permiten controlar remotamente los robots mencionados.

#### **3.1 Robot Rover 4WD**

El robot Rover 4WD, emula un vehículo de 4 ruedas, que recibe, a través del Cloud, las acciones que debe ejecutar. Para la confección de tal robot, se han utilizado los componentes siguientes:

##### ▪ **Chasis Rover de 4 ruedas**

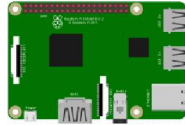
Consiste en dos placas de acrílico, con cuatro motores DC de 3v (dos del lado derecho: R1, R2, y dos del izquierdo: L1, L2), cuatro ruedas y un codificador de velocidad.



*Figura 2: Chasis Rover de 4 ruedas.*

- **Placa de desarrollo Raspberry**

Placa Raspberry Pi 3 con conexión WiFi, almacenamiento en memoria MicroSD de 32Gb y alimentación de energía de 5v.



*Figura 3: Placa Raspberry Pi 3.*

- **Placa de desarrollo Arduino**

Placa Arduino UNO, con sistema de alimentación de energía de 5v. Esta placa gestiona los sensores ultrasónicos para detectar obstáculos.



*Figura 4: Placa Arduino UNO.*

- **Puente H**

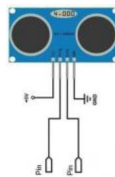
Módulo L298 para controlar la dirección de los motores; por medio de PWM, la velocidad de los mismos.



*Figura 5: Puente H Módulo L298.*

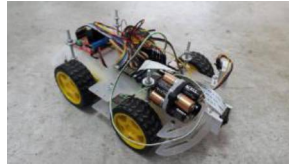
- **Sensor ultrasónico HC-SR04**

Detector de proximidad o detector de obstáculos. Cuantifica el delay de la señal, es decir, el tiempo que demora la señal en regresar el eco.



*Figura 6: Sensor ultrasónico HC-SR04.*

- **Prototipo Rover 4WD ensamblado**



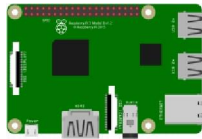
*Figura 7: Robot Rover 4WD.*

### 3.2 Cuadricóptero

El prototipo del robot cuadricóptero está integrado por una placa de desarrollo Raspberry Pi 3 y un Drone Parrot Bebop. El cuadricóptero está conectado vía Ethernet al Cloud de AWS por medio de la placa de desarrollo Raspberry Pi 3. La conexión entre la placa de desarrollo Raspberry y el Drone se realiza vía WiFi, estableciendo una conexión punto a punto segura, evitando así perder el control del dron.

- **Placa de desarrollo Raspberry**

Placa Raspberry Pi 3 con conexión WiFi, almacenamiento en memoria MicroSD de 16Gb y alimentación de energía de 5v.



*Figura 9: Placa Raspberry Pi 3.*

- **Drone Parrot Bebop**

Cuadricóptero de uso civil, de origen Francés, ligero, compacto, con cámara de video Full HD y autonomía de vuelo de 25 minutos con 2 baterías de 1200 mAh.



*Figura 10: Drone Parrot Bebop.*

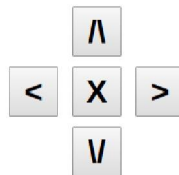
### 3.3 Instancia AWS EC2 – Aplicación para control remoto

Consiste en el despliegue de una instancia de AWS EC2 “t2.micro”, con 1 VCPU y 1GB de RAM; Sistema Operativo Linux distribución Ubuntu Server 16.04 LTS [15] [16]. Dicha instancia provee una interfaz WEB, desarrollada con HTML + JavaScript + Python, con el fin de controlar remotamente al sistema de multi-robots.

## 4 Trabajo experimental

Para el trabajo de investigación se requirió desarrollar una plataforma web que permita controlar remotamente, vía internet, el sistema de multi-robot que emula un vehículo de 4 ruedas y un cuadricóptero, no tripulados.

En la instancia de AWS, se configuró una interfaz WEB accesible por medio de su IP pública, lo que permite al usuario controlar el sistema multi-robot. Para el robot “Rover 4WD” se dispone de un teclado visual que permite gestionar las acciones sobre el mismo, por medio de los botones visuales o las teclas del teclado de una PC/Notebook: “arriba, abajo, izquierda y derecha” o con las teclas “A, W, S, D”; con este último caso, también se brinda la posibilidad de hacer girar al robot sobre su propio eje sin avanzar, esto se efectúa presionando las teclas “Q” o “E”. El uso de esta plataforma es similar al de un joystick, por ejemplo, para avanzar se mantiene presionada la tecla “arriba” (o la tecla “W”) y cuando se libera la presión de la misma, automáticamente el robot frena. Este algoritmo está desarrollado en JavaScript, integrado en la página HTML que ofrece la interfaz WEB. En segundo plano, la instancia de AWS envía los comandos al robot mediante el protocolo MQTT, publicando mensajes en el tópico “ROVER”, haciendo uso de la aplicación SDK de AWS IoT, implementada con Python. En la figura siguiente, se puede observar la interfaz WEB y la implementación del algoritmo:



**Figura 11:** Interfaz WEB - Robot “Rover 4WD”.

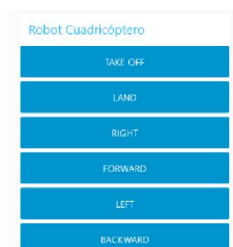
```
<html> <head>
<script      src="{{url_for('static',
filename='js/rover.js')}}"></script>
</head>
<body> <div align="center">
<div id="up">
<button id="bt_up">^</button> </div>
<div id="middle">
<button id="bt_left"><</button>
<button id="bt_stop">X</button>
<button id="bt_right">></button>
</div>
<div id="down">
<button id="bt_down">v</button> </div>
</div>
</body> </html>
```

**Figura 12:** Algoritmo - Robot “Rover 4WD”.

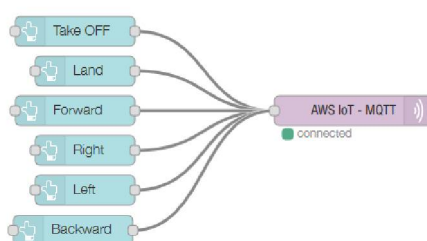
Para el caso del robot “Cuadricóptero”, también se dispone de una interfaz WEB que ofrece un teclado visual que permite controlar las acciones del robot (Take Off, Land, Forward, Right, Left, Backward). Esta interfaz está implementada con node-red-dashboard [17], y en segundo plano, la instancia de AWS envía los comandos al



robot mediante el protocolo MQTT utilizando node-red-mqtt-topic, publicando los mensajes en el tópico “DRONE”. En la figura siguiente, se puede observar la interfaz WEB y la implementación de su algoritmo:



**Figura 13:** Interfaz WEB - “Cuadricóptero”.



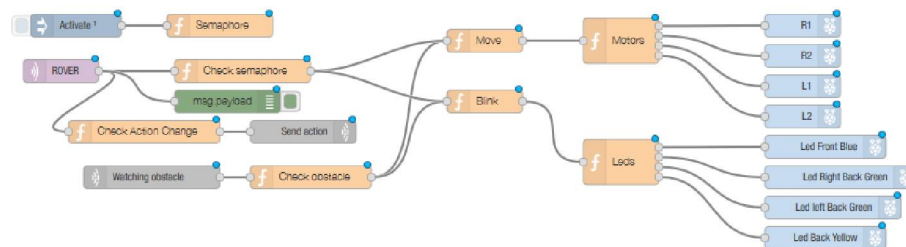
**Figura 14:** Algoritmo - “Cuadricóptero”.

El robot “Rover 4WD”, dispone de dos placas de desarrollo: Raspberry Pi3 y Arduino UNO. La placa Raspberry Pi3, vía la conexión WiFi, se encarga de recibir los comandos enviados por el protocolo MQTT (subscripto al tópico “ROVER”), activando o desactivando los motores (R1, R2, L1, L2); la lógica de este algoritmo se encuentra implementada con un flujo de Node-Red. Además, consta de dos algoritmos, los cuales reciben y envían datos al Arduino UNO respectivamente. El algoritmo que consume datos del Arduino UNO, recibirá una notificación cuando el robot esté por colisionar y, en caso afirmativo, luego recibirá otra notificación cuando el robot ya no esté más en peligro, ya sea porque el obstáculo se movió o porque se cambió la dirección del robot. Esta notificación se envía a la aplicación Node-Red para que realice las acciones correspondientes a través de un socket local en la placa Raspberry Pi3.

Por otro lado, la placa de desarrollo Arduino UNO se encarga de controlar que el robot no colisione, haciendo uso de los sensores ultrasónicos. Para determinar qué sensor ultrasónico utilizar, se reciben datos de la placa Raspberry Pi3, con el fin de conocer la orientación y dirección en la que viaja el robot y así evaluar el lado en el que se deben comprobar las distancias con los obstáculos. Cuando se está por colisionar, se envía una señal a la Raspberry Pi3 para que detenga los motores; una vez que se evade el obstáculo, ya sea porque el mismo se movió y no hay más peligro o, porque se cambió la dirección del robot a una en donde no hay peligro de colisión, se envía una señal afirmativa para que pueda avanzar.

En la figura siguiente, se puede observar el flujo de Node-RED para el robot “Rover 4WD”:

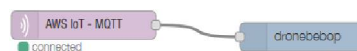




**Figura 15:** Flujo de Node-RED del robot “Rover 4WD”.

El robot “Cuadrícóptero”, dispone de una placa de desarrollo Raspberry Pi3 y un dispositivo de vuelo Drone Parrot Bebop. La placa Raspberry Pi3 establece una conexión punto a punto con el Drone Bebop, de forma segura, y vía conexión Ethernet en la placa, recibe los comandos enviados por el protocolo MQTT (subscripto al tópic “DRONE”); por cada comando recibido, se envía una orden específica al Drone Bebop, por medio del nodo de node-red-contrib-bebop2 [18].

En la figura siguiente, se puede observar el flujo de Node-RED para el robot “Cuadrícóptero”.



**Figura 16:** Flujo de Node-RED del robot “Cuadrícóptero”.

## 5 Conclusiones y Líneas de Trabajo Futuro

En esta investigación se ha implementado un sistema multi-robot conectado al Cloud. Se han diseñado los prototipos para simular un vehículo de 4 ruedas y un dispositivo de vuelo, no tripulados. Se han desarrollado los algoritmos que permitan a un usuario controlar remotamente los robots por medio de una interfaz WEB. Podemos concluir y reafirmar que Cloud Robotics es una tecnología que favorecerá el avance de los sistemas multi-robots reduciendo las limitaciones actuales.

Las líneas de trabajo a futuro relacionadas con este trabajo, incluyen la integración de una cámara al frente del robot “Rover 4WD” y la visualización de los videos en la interfaz WEB; así como también ampliar el espectro de visión de obstáculos adicionando más sensores ultrasónicos al chasis del robot.

Del mismo modo, es deseable ampliar el desarrollo de los algoritmos del nodo node-red-contrib-bebop2 para poder acceder a los datos que pueda proporcionar el robot “Cuadrícóptero”, como ser: video de la cámara incorporada, datos de vuelo, coordenadas de GPS, etc.

Por último, se espera adicionar uno o más robots Rover 4WD, adaptando la solución para simular una ciudad en la cual los sistemas multi-robot puedan trabajar colaborativamente.

## Referencias

1. Mell P., Grance T.: "The NIST Definition of Cloud Computing". Publicación Especial 800-145. Septiembre, 2011.
2. Pettoruti J., Rodríguez I., Chichizola F., De Giusti A.: "Análisis de la degradación de las comunicaciones en algoritmos de cómputo científico en un Cloud privado". XII Workshop de Procesamiento Distribuido y Paralelo (WPDP) – XVIII Congreso Argentino de Ciencias de la Computación (CACIC2012). Bahía Blanca, Argentina, 2012.
3. Rodríguez, I., Pettoruti, J.E., Chichizola, F., De Giusti, A.: Despliegue de un Cloud Privado para entornos de cómputo científico. Proceedings del XI Workshop de Procesamiento Distribuido y Paralelo (WPDP) - XVII Congreso Argentino de Ciencias de la Computación (CACIC 2011). La Plata, Argentina. 2011.
4. RoboEarth. In: <http://www.roboearth.org>.
5. Kuffner J.: "Cloud-enabled robots". IEEE-RAS International Conference on Humanoid Robot. Nashville, USA, 2010.
6. Wang L., Liu M., Meng M., Siegart R.: "Towards Real-Time Multi-Sensor Information Retrieval in Cloud Robotic System". IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI). Hamburgo, Alemania, 2012.
7. Turnbull L., et al.: "Cloud Robotics: Formation Control of a Multi Robot System Utilizing Cloud Infrastructure". Proceedings of IEEE – Southeastcon. Jacksonville, USA, 2013.
8. AWS IoT, <https://aws.amazon.com/es/iot>.
9. Bermudez I., Traverso S., Mellia M., Munafò M.: "Exploring the cloud from passive measurements: The Amazon AWS case". Proceedings of IEEE – INFOCOM, 2013.
10. Atzori L., Iera A., Morabito G.: "The Internet of Things: A survey". ELSEVIER Journal Computer Networks. Volume 54, Issue 15. 2010.
11. Protocolo MQTT, <http://mqtt.org>.
12. OASIS MQTT, <https://www.oasis-open.org>.
13. Informe técnico: AWS Whitepapers. "Core Tenets of IoT", <https://aws.amazon.com/es/whitepapers/>.
14. Node-RED, <https://nodered.org>.
15. AWS Instance Types: <https://aws.amazon.com/es/ec2/instance-types>.
16. AWS Marketplace: <https://aws.amazon.com/marketplace/pp/B00JV9JBDS>.
17. Node-Red-Dashboard, <https://github.com/node-red/node-red-dashboard>.
18. Node-Red-Contrib-Bebop2, <https://www.npmjs.com/package/node-red-contrib-bebop2>.